How incoherent can we be? Phase-encoded linearised inversion with random boundaries

Chris Leader and Ali Almomin

ABSTRACT

To perform linearised inversion on seismic exploration scale datasets we are continually looking for methods to accelerate computation and reduce data handling overhead. One option to accelerate reverse time imaging is to use random domain boundaries for the source wavefield computation, alleviating much of the required IO in favour of some additional computation. Additionally, data handling problems can be addressed by phase-encoding data (weighting, shifting, summing) and then inverting for a common model between realisations. Both random boundary and phase-encoding methods rely on wavefield incoherency during correlation and stacking to build a clean image. Here we investigate if these can be effectively used together, or if these techniques combined create wavefields that are too incoherent, slowing convergence as a function of cost when compared to linearised inversion without phase-encoding. We show that by using multiple realisations per iteration we can improve convergence and create cleaner reflectivity images.

INTRODUCTION

Reverse time migration (RTM) can provide accurate subsurface images because it applies the full, two-way wave equation. Thus steep dips, multiples, and prismatic waves can be imaged. However, RTM is the adjoint of an idealised modelling operator and not a full inverse operation, meaning that images can suffer from artefacts such as acquisition footprints, low-frequency noise and decreased resolution. We can approximate the inverse of this modelling procedure by using iterative least-squares inversion. However this can quickly become prohibitively expensive as each iteration is roughly twice the cost of a single migration. Furthermore, the formulation of RTM (the adjoint procedure) requires the source-side wavefield to be modelled, reversed, and sequentially correlated with the receiver-side wavefield. This modelling and reversal provides computational difficulties, as we have to save and re-inject a 4D source wavefield when applying 3D RTM. These two problems - inversion cost and source wavefield time reversal, - can be solved by taking advantage of correlation attributes and data redundancy.

When modelling the source wavefield we have to use a finite computational domain. This domain creates artificial boundary reflections, which must be removed else they create high-amplitude, coherent artefacts within the image. However, removing

these boundary reflections causes the modelling to be non-reversible, thus the entire wavefield must be saved and reused when correlating with the recorded data. We can use random boundaries (Clapp (2009); Fletcher and Robertsson (2011); Shen and Clapp (2011)) to make this computation time-reversible. By only saving the final wavefield snapshots we can now back propagate the wavefield to within numerical accuracy. Provided that boundary reflections are sufficiently incoherent, the RTM imaging condition and subsequent stacking over shots can reduce any residual incoherent noise to an imperctible level. Such a method is particularly useful for GPU computing. Here data must be read from disk, to the CPU, and then sent to the GPU, compounding any disk access. By removing the need for disk-saved source wavefields we accelerate GPU based RTM significantly.

We can address the inversion cost in a slightly different way. One method is to reduce the data size that we are imaging by combining sources. This can be done by shifting, weighting and summing shots to create one or several 'super shots' (Morton and Ober (1998); Romero et al. (2000)). The weights and shifts that we apply to individual shots are referred do as the encoding. Such a method can also be used for full waveform inversion (Gao et al. (2010);Krebs et al. (2009)). We can either combine sources into a single super shot or several super shots. When combined into one super shot the inversion is now independent of the number of sources, reducing the inversion cost by roughly this number (assuming full aperture for all shots). However, many crosstalk artefacts are seen when wavefields from different source experiments correlate coherently. These are slowly reduced when iterating, but by changing the encoding between iterations these are suppressed much faster. Romero et al. (2000) and Krebs et al. (2009) show that by using a single sample random encoding the best convergence rates are seen. The caveat of such a scheme is that we must recalculate the initial residual each time (since we have changed our observed data), making this method about 1.5 times more expensive.

Both random boundaries and phase-encoding can be very effective in accelerating linearised inversion. However, they both introduce a considerable amount of noise into the system and rely on correlation, stacking and inversion to reduce this. By combining these methods we may be making the system too incoherent, slowing the inversion process down as a function of cost. Here we investigate convergence properties of these techniques and how we can try to create cleaner gradients within each iteration.

INVERSION WITH RANDOM BOUNDARIES

Under a single Born scattering assumption we can describe the recorded data as

$$d(\mathbf{x}_r, \mathbf{x}_s, \omega) = \sum_{\mathbf{x}_s, \omega} f(\omega) G_0(\mathbf{x}, \mathbf{x}_s, \omega) m(\mathbf{x}) \sum_{\mathbf{x}_r} G_0(\mathbf{x}, \mathbf{x}_r, \omega). \tag{1}$$

The adjoint process can thus be written as

$$m(\mathbf{x}) = \sum_{\mathbf{x}_s,\omega} f(\omega) G_0(\mathbf{x}, \mathbf{x}_s, \omega) \sum_{\mathbf{x}_r} G_0(\mathbf{x}, \mathbf{x}_r, \omega) d^*(\mathbf{x}_r, \mathbf{x}_s, \omega). \tag{2}$$

Where d is the data, G_0 are the respective Green's functions, m the model, $\mathbf{x}$ the three-dimensional model coordinate, $\mathbf{x}_{r,s}$ the three-dimensional source and receiver coordinates, ω temporal frequency, $f(\omega)$ the source waveform and $*$ denotes the complex conjugate. It is this complex conjugate that reverses the sense of time, meaning that in the time domain we back propagate the source and receiver wavefields and correlate them at each imaging time step. This complex conjugate and the subsequent source wavefield modelling create the need for either saving this wavefield or forming a time-reversible source wavefield. Random boundaries scatter this wavefield incoherently whilst adhering to the conservation of energy; after propagation correlation and stacking will reduce residual noise.

Random boundary noise in an RTM image is stacked out at a rate of $\sqrt{N}$, where N is the number of sources; often we see better performance than this. By using a different random boundary for each shot experiment we see random noise levels at an acceptable level after combining around 50 shots. Some effects of this can be seen below, where we use a GPU based RTM algorithm and model and migrate 100 shots over a section of the SEAM velocity model (Figure 1), simulating a marine survey. We then perform 5 iterations of least-squares linearised inversion with the same dataset. Here we have 50 inline shots at 100m spacing, 2 crossline shots with 1km spacing and 825x200 receivers. Comparison of such a scheme compared to source saving are shown in Clapp (2009).

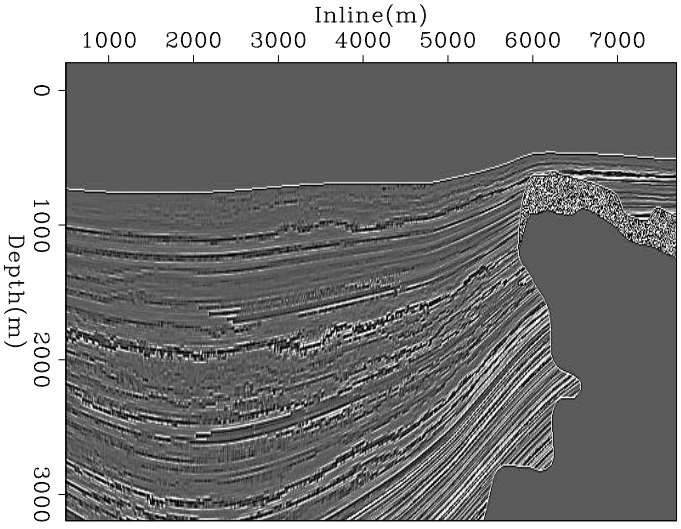

Figure 1: A 2D slice from the reflectivity model that we are attempting to recover. [**ER**]

As expected, in Figure 2 we see image quality improve when extending this procedure to an iterative least squares inversion. The footprint from the limited acquisition begins to dissipate and we see generally higher frequency content and more balanced, geologically accurate amplitudes. However, since only a limited number of iterations

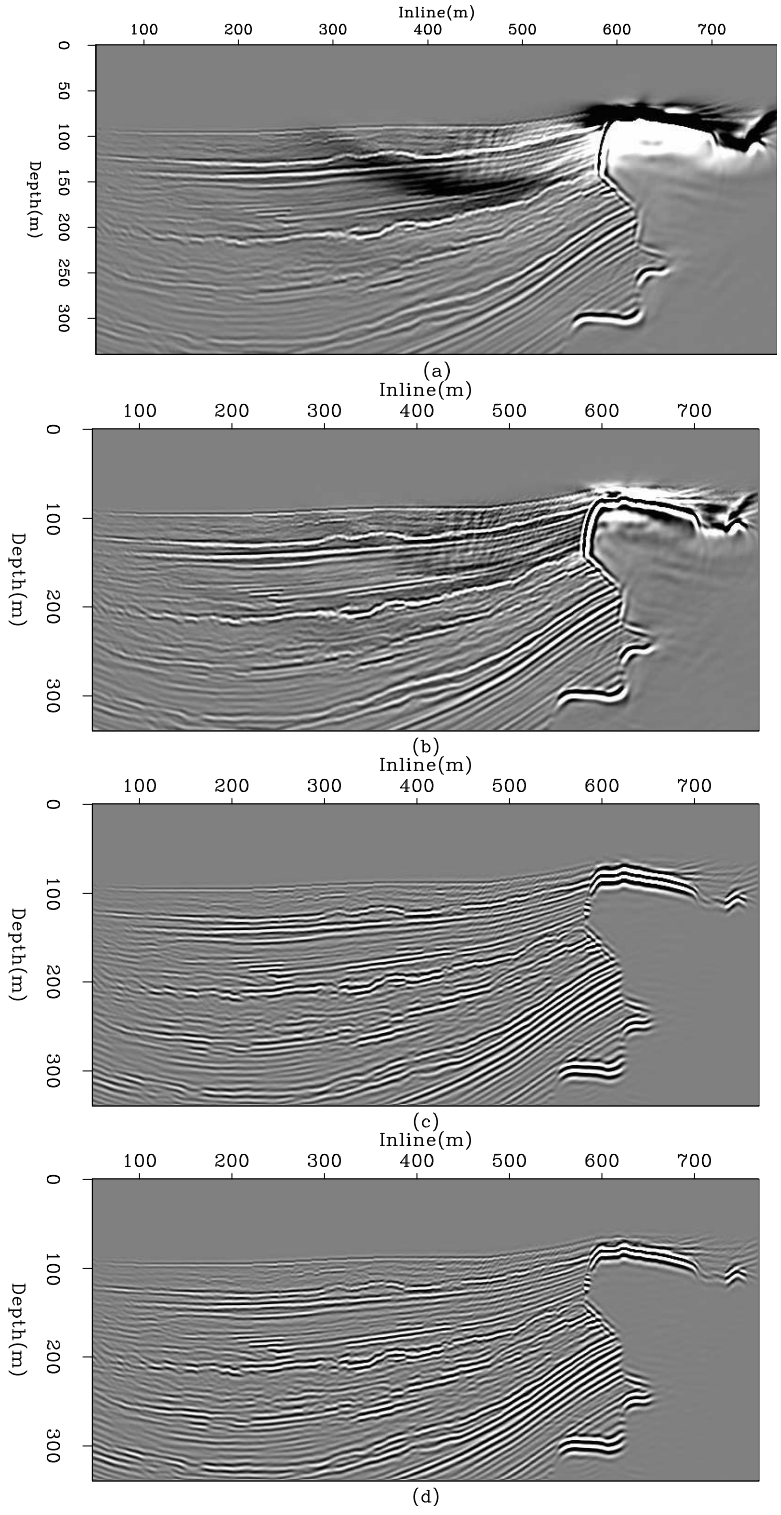

Figure 2: RTM and linearised inversion example 2D slices. (a) shows the raw RTM result, (b) raw inversion after 5 iterations, (c) is (a) after a lowcut wavenumber filter and (d) is (b) after the same lowcut filter. [**CR**]

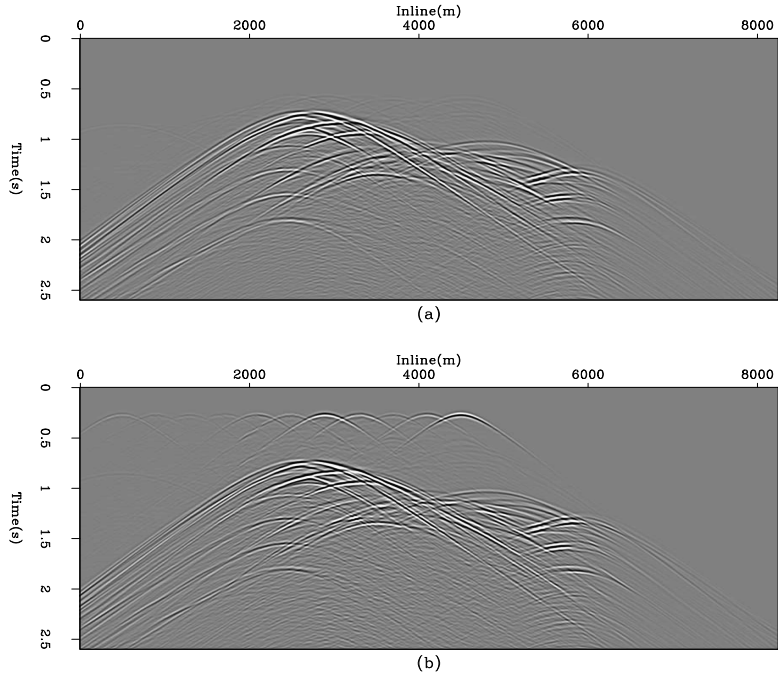

Figure 3: An example of source self-correlation. a) shows the real data after encoding 10 shots. b) shows the modelled data when the water column noise has not been muted. Artifacts similar to direct arrivals not present in a) can be seen. [**CR**]

have been performed some artefacts remain, from both the random boundaries and from the inverse system.

When blindly extending this process to an inversion, relatively poor convergence can be seen, especially with respect to the data space residual. The reason for this can be seen when looking at this residual. The random boundary image (the gradient, in this case) features noise in the water column and a source location imprint artefact. When modelling data over this image we see an artefact manifested as a direct arrival from the source self-correlating. This can be removed by either muting the water column in the image or by time muting the remodelled data before back-propagation. An example of this under a phase-encoded setting is shown in Figure 3. Furthermore, we can improve convergence by changing the random boundaries as a function of iteration number, as well as a function of shot position.

The computational advantage for GPU based RTM is considerable. We can consider three scenarios - when the source wavefield must be entirely saved to disk, when the source wavefield can held in the CPU memory and when using random boundaries. One should note the last of these requires an extra computation - the source wavefield back propagation during RTM. In total computation time asynchrous disk wavefield transfer is nine times slower than random boundary RTM, and if the source wavefield can be compressed and stored on the CPU memory this is 1.5 times slower. Of course, these conclusions are strongly related to the speed of the disks being used.

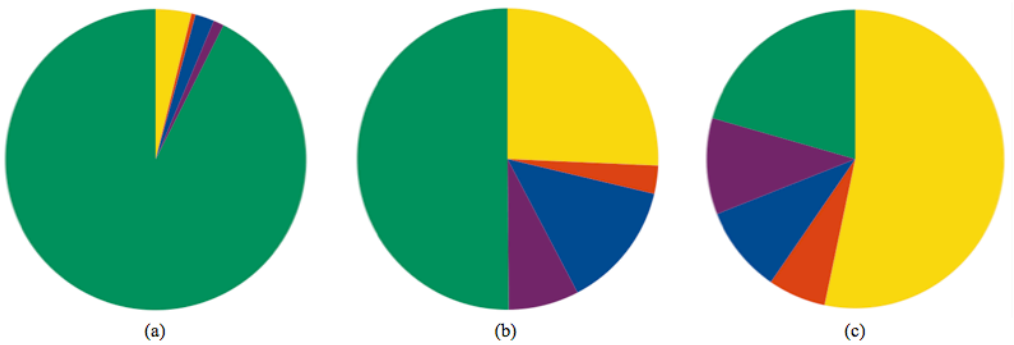

Figure 4: Compute time comparisons for three RTM regimes. Green denotes IO, yellow propagation, blue damping, red injection and purple imaging. (a) uses asynchronous disk transfers, (b) assumes the entire source wavefield can be held in CPU memory and (c) uses random boundaries. [**NR**]

PHASE ENCODING WITH RANDOM BOUNDARIES

Phase encoding can have multiple meanings; herein we describe the process of weighting and combining shots to reduce data size. We do this by creating a matrix of weights and applying it to our data.

$$\widetilde{d}(\mathbf{x}_r, p_s, \omega) = \sum_{\mathbf{x}_s} \alpha(\mathbf{x}_s, p_s) d(\mathbf{x}_r, \mathbf{x}_s, \omega) \tag{3}$$

Now $\widetilde{d}$ denotes our phase encoded data, α is a sequence randomly selecting either 1 and -1, and p_s is some sort of realisation index. When combining all shots together p_s will be 1. For the forward process we propagate a source function encoded with the same sequence α.

Augmenting random boundary linearised inversion with phase encoding requires some additional thought. In the case where we combine all shots to one super shot, we are now propagating 100 weighted shots through the same random boundary. On a shot by shot basis this is acceptable, as each wavefield will be incident on the boundary at a different angle and hence scatter differently. However, when performing this conventionally we formulate a gradient for each shot separately and then sum them to create our final gradient, reducing our noise by $\sqrt{N}$. Furthermore, there are only two coherent wavefields (the source and receiver) that can correlate with the scattered field to induce noise. When combining all shots together, we do not quite see this behaviour because we now have every scattered field (unique per shot) correlating with every scattered field and with every coherent field, of which there are now $2N$. Fortunately, the correlation of scattered fields should also be a random walk reduction at $\sqrt{N}$. Typically with phase encoded linearised inversion after *nshots* iterations one expects to see an acceptably clean image. When including random boundaries we see a noisier image, as expected, but all key features are present. Images can still

be artefact-laden, especially in areas of low illumination, but this is typical of both random boundary and phase-encoded imaging when done independently.

We have several options to mitigate this. The best results are seen by combining several random subsets of shots, calculating a gradient for each subset and combining these to form the gradient for a single iteration. Whilst this appears more computationally demanding, the fact that single super-shot GPU based inversion is not further parallelisable over shots makes this approach seem more appealing. Over a node containing 8 Fermi M2090 cards we can create the gradient for 8 different super shots and combine these for little extra time cost, giving a cleaner gradient per iteration. We now see more favourable convergence characteristics and slightly cleaner images. With naive phase encoded inversion we see a convergence to 60% within the data-space residual norm after 100 iterations with 50 combined shots. With a water column mute applied we see convergence to 56% and with multiple, stacked realisations we see convergence to 53%. Once the data error is below 50% we tend to see very gradual improvements for all situations. One full, non-encoded iteration with random boundaries (roughly the same cost) takes us to 74% data error.

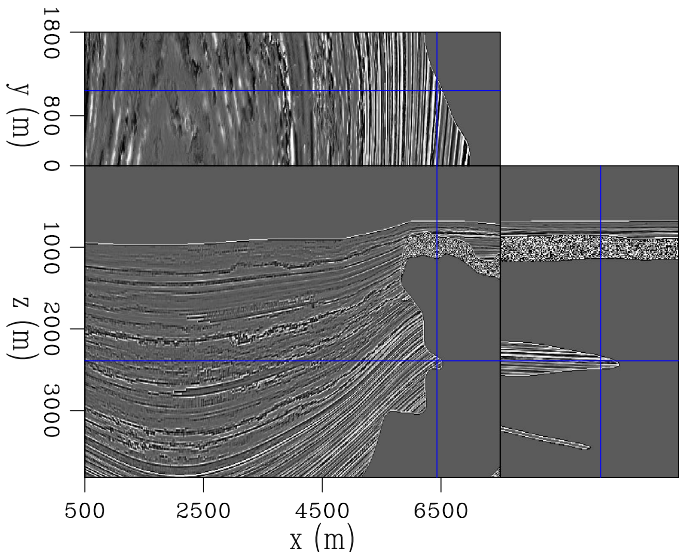

Figure 5: The reflectivity model that we are inverting for. [**ER**]

Figure 6 compares the results of conventional, separated linearised inversion with phase encoded inversion, over the 3D model shown in Figure 5. Here we had 120 shots in total, 60 inline at 100m spacing and two crossline at 1km spacing. Again, receivers were in a 825x200 grid. Images (a) and (c) are equivalent in computational cost, as are images (b) and (d). The first noticeable aspect are the low-frequency artefacts present in the non-encoded image. These occur due to wavefields moving in the same direction correlating and are prevalent over high-reflectivity, high-contrast features such as salt boundaries. Often the first several iterations of linearised inversion will work on removing these artefacts before focusing on other areas, as can also be seen in Figure 2. The second noticeable aspect is that the frequency content of the phase encoded image is much higher. The resolution, especially on the salt edges, is greater; high-frequency noise is also present, but this is expected. The additional iterations

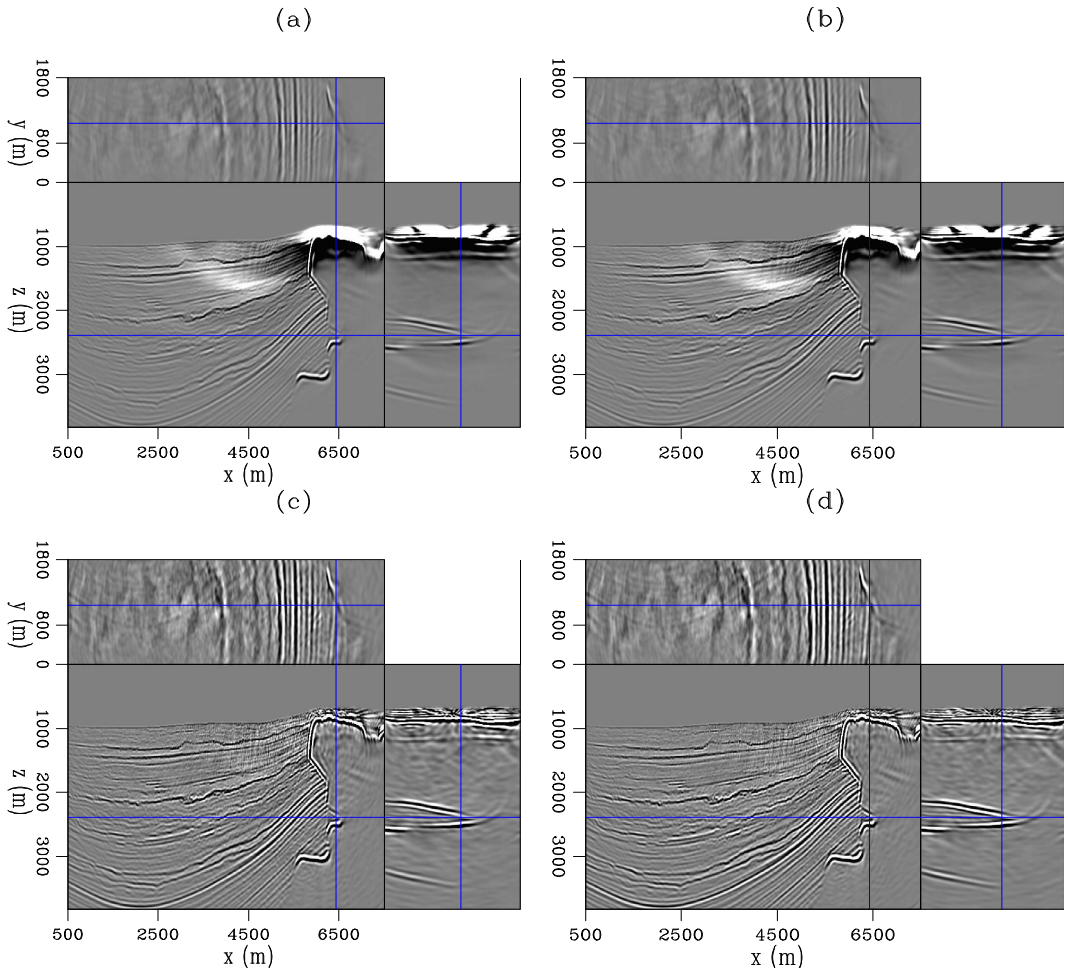

Figure 6: Linearised inversion examples. (a) shows the model after one iteration of conventional inversion, (b) the model after two iterations, (c) phase encoded inversion after 80 iterations - equivalent cost to (a) and (d) phase encoded inversion after 160 iterations - equivalent cost to (b). [**CR**]

performed here have successfully begun to remove the effect of the source wavelet in the image, whereas in the RTM image this wavelet is squared. Figure 7 shows these same images after a low-cut bandpass filter and some light amplitude gain. The filter has removed the low-wavenumber noise, but at the expense of the vertical salt edge. The conventional images look much improved, but the resolution and general content of the phase-encoded images still seem preferable.

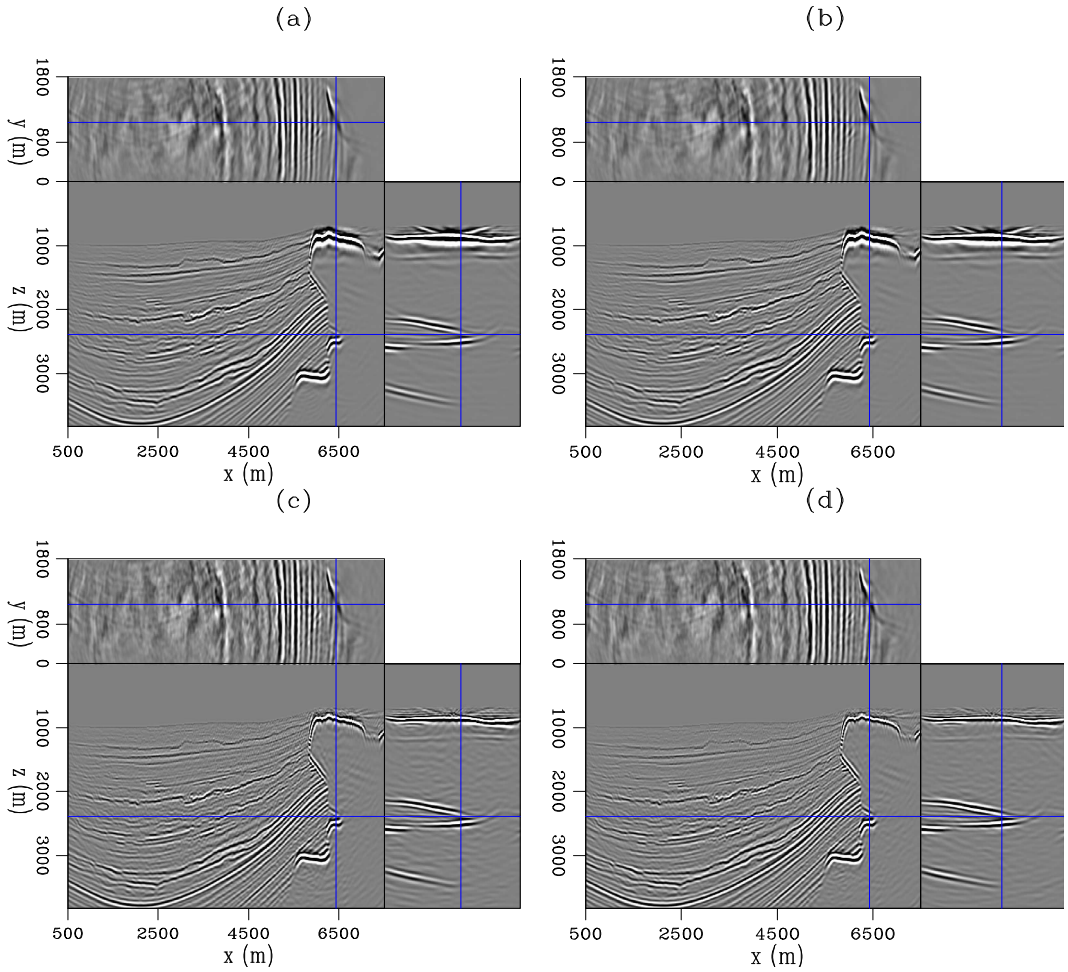

Figure 7: The same set of results as in Figure 6 but each with a low-cut spatial frequency filter and some amplitude gain. [**CR**]

Figures 6 and 7 show that for equivalent cost, even when augmented with random boundaries, phase encoded linearised inversion can yield high quality images. This scheme is also incredibly well-adapted for GPU computing - there is no IO during propagation in the forward or adjoint scheme, and the objects we need to copy to the GPU are all the size of the model, the size of one shot, or smaller.

Residual behaviour with iteration number can be seen in Figure 8. As a reference point for separated inversion after two iterations the respective normalised residual difference norms (normalised to 100) were 88.9 and 79.9, which are not significantly smaller than those shown. However, in terms of cost the former of these would

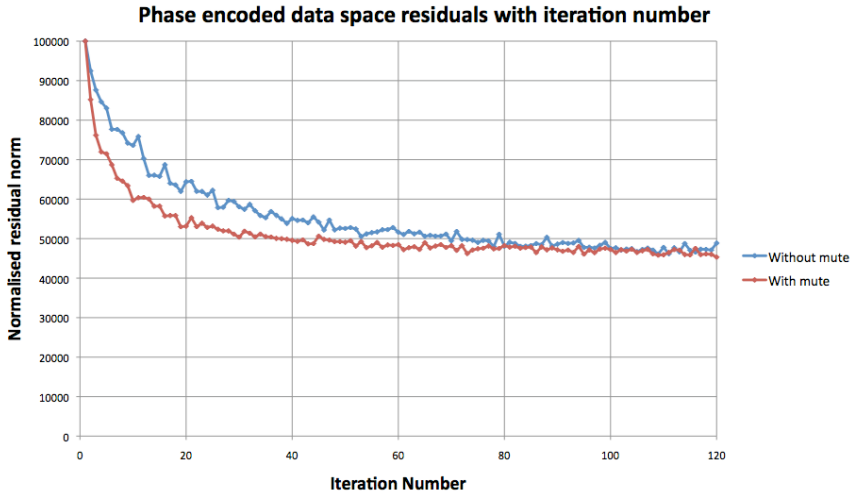

Figure 8: Normalised residual evolution as a function of iteration number. [**NR**]

appear at 80 iterations, the latter at 160. With this is mind we see that our phase-encoding scheme is doing a vastly more efficient job at data fitting, in an l_2 sense. However, when interpreting these scalar fits we must consider that the l_2 residual norm does not well represent high-frequency noise, and so two images with an apparently similar residual may have quite different high-frequency noise characteristics. It is due to this that we must take the phase encoded scheme to so many iterations. Typically, we will need at least $nshots/2$ if not $nshots$ iterations for a clean image here. With conventional inversion we need 5-10 iterations to remove the low-frequency salt artefacts and many more for amplitude and acquisition imbalances.

CONCLUSIONS

Both phase encoding and random boundary propagation can be very effective in accelerating linearised inversion. We presented the individual benefits and how combining these methods can lead to a powerful inverse scheme with reference to inverse imaging on GPUs. We can conclude that these techniques can be used together, and acceptable images are obtained within the cost of a conventional RTM migration. However, to improve convergence we can see that a mute must be applied to avoid direct-arrival type artefacts in the Born-modelled data, and that we can improve convergence again by stacking separate gradient realisations per iteration. For future work we will explore the best method of cleaning the gradients between iterations and will update the non-linear solver to see if we can improve our convergence rate once we are within 50% data error.

ACKNOWLEDGMENTS

The author would like to thank NVIDIA for their continued support with troubleshooting our GPU cards.

REFERENCES

Clapp, R. G., 2009, Reverse time migration with random boundaries: SEG Technical Program Expanded Abstracts, **28**, 2809–2813.

Fletcher, R. P. and J. O. A. Robertsson, 2011, Time-varying boundary conditions in simulation of seismic wave propagation: SEG Technical Program Expanded Abstracts, **30**, 2957–2961.

Gao, F., A. Atle, and P. Williamson, 2010, Full waveform inversion using deterministic source encoding: SEG Technical Program Expanded Abstracts, **29**, 1013–1017.

Krebs, J. R., J. E. Anderson, D. Hinkley, R. Neelamani, S. Lee, A. Baumstein, and M.-D. Lacasse, 2009, Fast full-wavefield seismic inversion using encoded sources: Geophysics, **74**, WCC177–WCC188.

Morton, S. A. and C. C. Ober, 1998, Fastershot-record depth migrations using phase encoding: SEG Technical Program Expanded Abstracts, **17**, 1131–1134.

Romero, L. A., D. C. Ghiglia, C. C. Ober, and S. A. Morton, 2000, Phase encoding of shot records in prestack migration: Geophysics, **65**, 426–436.

Shen, X. and R. G. Clapp, 2011, Random boundary condition for low-frequency wave propagation: SEG Technical Program Expanded Abstracts, **30**, 2962–2965.